Contents

Install with Helm

5 minute read ✔ page test

Follow this guide to install and configure an Istio mesh using Helm for in-depth evaluation.

The Helm charts used in this guide are the same underlying charts used when installing Istio via Istioctl or the Operator.

This feature is currently considered alpha.

Prior to Istio 1.9.0, installations using the Helm charts required hub and tag arguments: `--set global.hub="docker.io/istio"` and `--set global.tag="1.8.2"`. As of Istio 1.9.0 these are no longer required.

Prerequisites

1. Download the Istio release.
2. Perform any necessary platform-specific setup.
3. Check the Requirements for Pods and Services.
4. Install a Helm client with a version higher than 3.1.1.

Helm 2 is not supported for installing Istio.

The commands in this guide use the Helm charts that are included in the Istio release package located at `manifests/charts`.

Installation steps

Change directory to the root of the release package and then follow the instructions below.

> The default chart configuration uses the secure third party tokens for the service account token projections used by Istio proxies to authenticate with the Istio control plane. Before proceeding to install any of the charts below, you should verify if third party tokens are enabled in your cluster by

following the steps describe here. If third party tokens are not enabled, you should add the option `--set global.jwtPolicy=first-party-jwt` to the Helm install commands. If the `jwtPolicy` is not set correctly, pods associated with `istiod`, gateways or workloads with injected Envoy proxies will not get deployed due to the missing `istio-token` volume.

1. Create a namespace `istio-system` for Istio components:

```
$ kubectl create namespace istio-system
```

2. Install the Istio base chart which contains cluster-wide resources used by the Istio control plane:

```
$ helm install istio-base manifests/charts/base -n istio-sy
stem
```

3. Install the Istio discovery chart which deploys the `istiod` service:

```
$ helm install istiod manifests/charts/istio-control/istio-
discovery \
    -n istio-system
```

4. (Optional) Install the Istio ingress gateway chart which contains the ingress gateway components:

```
$ helm install istio-ingress manifests/charts/gateways/isti
o-ingress \
    -n istio-system
```

5. (Optional) Install the Istio egress gateway chart which contains the egress gateway components:

```
$ helm install istio-egress manifests/charts/gateways/istio
-egress \
    -n istio-system
```

Verifying the installation

Ensure all Kubernetes pods in `istio-system` namespace are deployed and have a `STATUS` of `Running`:

```
$ kubectl get pods -n istio-system
```

Updating your Istio configuration

You can provide override settings specific to any Istio Helm chart used above and follow the Helm upgrade workflow to customize your Istio mesh installation.

The available configurable options can be found by inspecting the top level `values.yaml` file associated with the Helm charts located at `manifests/charts` inside the Istio release package specific to your version.

Note that the Istio Helm chart values are under active development and considered experimental. Upgrading to newer versions of Istio can involve migrating your override values to follow the new API.

For customizations that are supported via both `ProxyConfig` and Helm values, using `ProxyConfig` is recommended because it provides schema validation while unstructured Helm values do not.

Create a backup

Before upgrading Istio in your cluster, we recommend creating a backup of your custom configurations, and restoring it from backup if necessary:

```
$ kubectl get istio-io --all-namespaces -oyaml > "$HOME"/istio_resource_backup.yaml
```

You can restore your custom configuration like this:

```
$ kubectl apply -f "$HOME"/istio_resource_backup.yaml
```

Migrating from non-Helm installations

If you’re migrating from a version of Istio installed using `istioctl` or Operator to Helm (Istio 1.5 or

earlier), you need to delete your current Istio control plane resources and re-install Istio using Helm as described above. When deleting your current Istio installation, you must not remove the Istio Custom Resource Definitions (CRDs) as that can lead to loss of your custom Istio resources.

It is highly recommended to take a backup of your Istio resources using steps described above before deleting current Istio installation in your cluster.

You can follow steps mentioned in the Istioctl uninstall guide or Operator uninstall guide depending upon your installation method.

Uninstall

You can uninstall Istio and its components by uninstalling the charts installed above.

1. List all the Istio charts installed in `istio-system` namespace:

```
$ helm ls -n istio-system
NAME            NAMESPACE       REVISION    UPDATED
                        STATUS      CHART
   APP VERSION
istio-base      istio-system    1           ... ... ... ...
                        deployed    base-1.9.0
istio-egress    istio-system    1           ... ... ... ...
                        deployed    istio-egress-1.9.0
istio-ingress   istio-system    1           ... ... ... ...
                        deployed    istio-ingress-1.9.0
istiod          istio-system    1           ... ... ... ...
                        deployed    istio-discovery-1.9.0
```

2. (Optional) Delete Istio ingress/egress chart:

```
$ helm delete istio-egress -n istio-system
$ helm delete istio-ingress -n istio-system
```

3. Delete Istio discovery chart:

```
$ helm delete istiod -n istio-system
```

4. Delete Istio base chart:

By design, deleting a chart via Helm doesn’t delete the installed Custom Resource Definitions (CRDs) installed via the chart.

```
$ helm delete istio-base -n istio-system
```

5. Delete the `istio-system` namespace:

```
$ kubectl delete namespace istio-system
```

Uninstall stable revision label resources

If you decide to continue using the old control plane, instead of completing the update, you can uninstall the newer revision and its tag by first issuing `helm template istiod manifests/charts/istio-control/istio-`

`discovery -s templates/revision-tags.yaml --set revisionTags={prod-canary} --set revision=canary -n istio-system | kubectl delete -f -`. You must them uninstall the revision of Istio that it pointed to by following the uninstall procedure above.

If you installed the gateway(s) for this revision using in-place upgrades, you must also reinstall the gateway(s) for the previous revision manually, Removing the previous revision and its tags will not automatically revert the previously in-place upgraded gateway(s).

(Optional) Deleting CRDs installed by Istio

Deleting CRDs permanently removes any Istio resources you have created in your cluster. To permanently delete Istio CRDs installed in your cluster:

```
$ kubectl get crd | grep --color=never 'istio.io' | awk '{print $1}' \
    | xargs -n1 kubectl delete crd
```